

Д. О. ЗОЛОТАРЬОВ

РОЗПОДІЛЕНА СИСТЕМА АВТОМАТИЗОВАНИХ ОБЧИСЛЕНЬ НА БАЗІ ХМАРНОЇ ІНФРАСТРУКТУРИ

Стаття присвячена теоретичному дослідженню та розробці розподіленої системи автоматизованих обчислень та аналізу на базі хмарної інфраструктури. **Предметом** дослідження є теоретичні та практичні засади побудови систем автоматизованих розрахунків та аналізу, що засновані на клієнт-серверній архітектурі у розподіленій інфраструктурі на базі хмарних технологій. **Метою** статті є розробка та обґрунтування практичних рекомендацій щодо формування інфраструктури системи автоматизованих обчислень, вибору її складових елементів та їх компонентів. **Завдання** роботи: виявити необхідні структурні елементи системи автоматизованих обчислень та надати для кожного з них аналіз складових компонентів та функціонального навантаження, поставити конкретні задачі для побудови кожного із них та обґрунтувати вибір інструментів для їх вирішення. У ході дослідження використано **методи** системного аналізу для декомпозиції складної системи на елементи та кожного елемента на функціональні компоненти. У результаті дослідження встановлено, що інфраструктура системи має складатися з: єдиних ініціатора обчислень та центра обчислень, каналів зв'язку із клієнтами, та кінцевими клієнтами, що обробляють результати. Уся система має працювати у реальному або близькому до реального часу, що має бути досягнуто через обробку подій операційної системи та зовнішніх сервісів. Канали зв'язку мають бути максимально універсальними та надавати широкі можливості клієнтам для доступу до даних. Кінцеві клієнти мають бути двох типів: повноцінні для гнучкої та індивідуальної обробки даних, та спрощені для інтегрованого середовища узагальненої обробки даних у мобільному або десктопному браузері. За отриманими рекомендаціями розроблений один із варіантів такої системи, показані принципи його роботи та наведені результати. **Висновки.** Розроблені теоретичні та практичні основи для реалізації систем розподілених автоматизованих обчислень та аналізу на базі хмарних технологій. Показано підвищення надійності збереження результатів моделювання у такій системі через їх потрібне дублювання завдяки: локальному сховищу, використанню зовнішніх бази даних та файлового сервісу. Надані доводи щодо підвищення зручності та гнучкості у обробці результатів через можливість використання сторонніх аналітичних додатків, що підтримують завантаження даних із таких джерел. Показана економічна вигода від використання описаної системи. Наведені майбутні шляхи її вдосконалення.

Ключові слова: хмарні технології; розподілена інфраструктура; хмарні автоматизовані обчислення; економія ресурсів та коштів; ітераційні алгоритми; Mathematica.

Вступ

Досягнення сучасної прикладної науки є значною мірою результатом роботи колективів вчених, дослідників та інженерів, часто міждисциплінарних. Приміром такої взаємодії можуть служити роботи поточного року [1-7], у кожній з яких приймала участь значна кількість спеціалістів, що належали не тільки до різних організацій, а й часто до різних напрямків досліджень. Наочно видно, що отримання цих результатів стало можливим тільки завдяки міжпрофесійній кооперації.

Особливістю такої взаємодії є те, що кожен використовує звичні у його області інструменти розробки, дослідження та аналізу, що важко поєднати у одну систему. Тому під час важливих або довготривалих досліджень у такій групі, щоб організувати роботу, розроблюються або впроваджуються спеціалізовані системи взаємодії для передачі результатів між учасниками. Ці системи часто створюються організаціями під конкретні задачі і тому для кожного випадку будуть відрізнятися в силу більшої різниці у задачах команд та їх структурі.

Крім того у сьогоденні умовах люди, що беруть безпосередню участь у командних дослідженнях, можуть бути як розподілені по світу, в тому числі за часовими поясами, так і знаходитись у одному місті але без можливості бути разом із іншими учасниками. Щоб організувати їх комфортну взаємодію також розроблюються аналогічні системи, що дозволяють обмінюватися результатами досліджень у автоматичному режимі у реальному або

близькому до реального часу.

Задачі, що стоять перед такими командами можуть мінятися, а з ними змінюється й розроблена система, що призводить до витрат коштів на невиробничі процеси.

Розвиток та вдосконалення хмарних IaaS технологій дозволяє підійти до вирішення цієї задачі з іншого боку – не купувати та володіти дорогою технікою для побудови інфраструктури та елементів системи, а орендувати її на потрібний час у потрібному об'ємі із потрібними ресурсами. У разі зміни потреб така інфраструктура і ресурси кожного елементу можуть бути змінені за лічені хвилини. Собівартість володіння нею є, вочевидь, значно меншою та приносить найбільшу економію коштів у разі частоті зміни задач, що спричиняють істотну зміну конфігурації інфраструктури, або при нерегулярному її використанні через можливість виключення як інфраструктури цілком так і окремих її елементів.

Увага цієї статті приділена проблемі побудови системи автоматизованих обчислень для випадку використання ітераційних алгоритмів програми [8-11] із одним ініціатором обчислень із використанням у якості обчислювального центру ядра математичного процесору Wolfram Mathematica [12] на базі хмарного IaaS сервісу, що враховує таку важливу складову для віддаленого виконання як автоматизований нагляд за процесом розрахунків із миттєвим сповіщенням та розраховану на використання максимально широкого спектру кінцевих клієнтів, кожен з яких повинен мати змогу автоматичної обробки результатів обчислення у близькому до

реального часу.

Метою цієї статті є розробка та обґрунтування практичних рекомендацій щодо формування інфраструктури такої системи, вибору її складових елементів та їх компонентів. Завдання – виявити необхідні її структурні елементи та надати для кожного з них аналіз складових компонентів та функціонального навантаження, поставити конкретні задачі для побудови кожного із них та обґрунтувати вибір інструментів для їх вирішення.

Ітераційні алгоритми обрані через те, що їх особливістю є "природна" здатність до збереження стану після закінчення кожної ітерації – результати усіх раніше обчислених ітерацій можна зберігати та завантажувати, тим самим повністю відновлювати збережений стан програми для певного кроку алгоритму. Неітераційні алгоритми, що здатні зберігати свій стан та результати проміжних обчислень для аналізу, також можуть бути використані.

Математичний процесор Wolfram Mathematica був обраний через внутрішню побудову за клієнт-серверною архітектурою: клієнт Mathematica та ядро WolframKernel, та платформонезалежність серверу.

Викладемо теоретичні основи для побудови такої системи та результат, що є одним із можливих варіантів її реалізації.

Складові системи автоматизованих обчислень

Для побудови системи автоматизованих обчислень ітераційними алгоритмами у Mathematica позначимо проблеми, що мають бути вирішені. Це будуть, по-перше, запуск на виконання програми у ядрі Mathematica єдиним ініціатором обчислень та автоматичний нагляд за коректною роботою ядра із сповіщенням ініціатора обчислень. По-друге, збереження та передача результатів до кінцевих клієнтів для обробки та автоматизація обробки на таких клієнтах.

Для розв'язання поставлених проблем сформулюємо задачі у вигляді характеристик елементів за направленнями.

Ініціатор обчислень має захищений мережевий доступ до центру обчислень, у автоматичному режимі зберігає документ Mathematica після кожного виводу проміжних результатів чи іншого.

Центр обчислень надає захищений мережевий доступ для ініціатора обчислень, має зовнішнє протоколювання роботи програми, миттєво сповіщає про збій у процесі обчислення, зберігає результати обчислення кожної ітерації, очищає займані ресурси комп'ютера після закінчення обчислень.

Канал зв'язку із клієнтами, що обробляють результати передає у близькому до реального часу результати обчислення кожної ітерації та протокол роботи програми до кінцевих клієнтів.

Клієнти обчислювальної системи, що обробляють дані у близькому до реального часу, мають можливість автоматичного аналізу у системі Mathematica, сторонніх додатках та платформонезалежному середовищі.

Обрання хмарних технологій для обчислень мають наступні переваги: такий сервер може бути швидко розгорнутий, швидко змінена його конфігурація на ту, що задовольняє потребам поточної задачі, швидко клонований для паралельного обчислення декількох задач. У якості потужностей був обраний один із світових лідерів хмарних технологій DigitalOcean, де були розташовані: сервер обчислень із ядром Wolfram Mathematica 11.3.0 for Linux x86 (64-bit) [12] на базі ОС Ubuntu 20.04 LTS x64, сервер баз даних MySQL 8.0.20 Community Server на тій самій ОС, веб-сервер на базі Apache 2.4.29 (Ubuntu) та фреймворку Laravel 6 [14] на тій самій ОС.

Блок-схема системи автоматизованих обчислень показана на рис. 1.

Робота системи починається із підключення клієнта Mathematica на комп'ютері, що ініціює обчислення, до сервера обчислень та запуск програми у ядрі Mathematica на виконання.

Фоновий процес наглядчика контролює роботу ядра та у разі його аварійної зупинки відправляє сповіщення на e-mail ініціатора обчислень та у приватну групу Slack, до якої мають доступ усі зацікавлені особи.

Після виконання обчислення кожної ітерації її результати зберігаються у спеціально виділений локальній директорії на сервері обчислень, а фоновий процес файлового наглядчика їх у реальному часі автоматично завантажує, по-перше, у спеціально виділену директорію на сервері Dropbox через web-API, по-друге, у виділену базу даних на MySQL-сервері. Аналогічним образом проходить завантаження протоколу роботи програми – лог-файлів.

Обробка та візуалізація процесу обчислень у близькому до реального часу на клієнтських комп'ютерах реалізується двома шляхами: по-перше, через клієнт Mathematica або інший аналітичний додаток, що обробляє файли, отримані із Dropbox у локальну директорію комп'ютера, по-друге, через браузер, що підключається до веб-серверу, що з'єднуються із сервером бази даних.

Через мульти-клієнтну архітектуру, що від самого початку присутня у веб-сервері, кількість клієнтів, що підключаються через браузер, обмежена лише потужністю сервера та може бути підвищена за потреби. Крім того, й кількість веб-серверів, що підключається до єдиної бази даних теж обмежується тільки потужністю останньої.

Для клієнтів, що під'єднуються через сервіс Dropbox, не обмежена не тільки кількість подібних комп'ютерів-клієнтів, але й кількість запущених додатків, що можуть оброблювати паралельно ті ж самі дані різними способами.

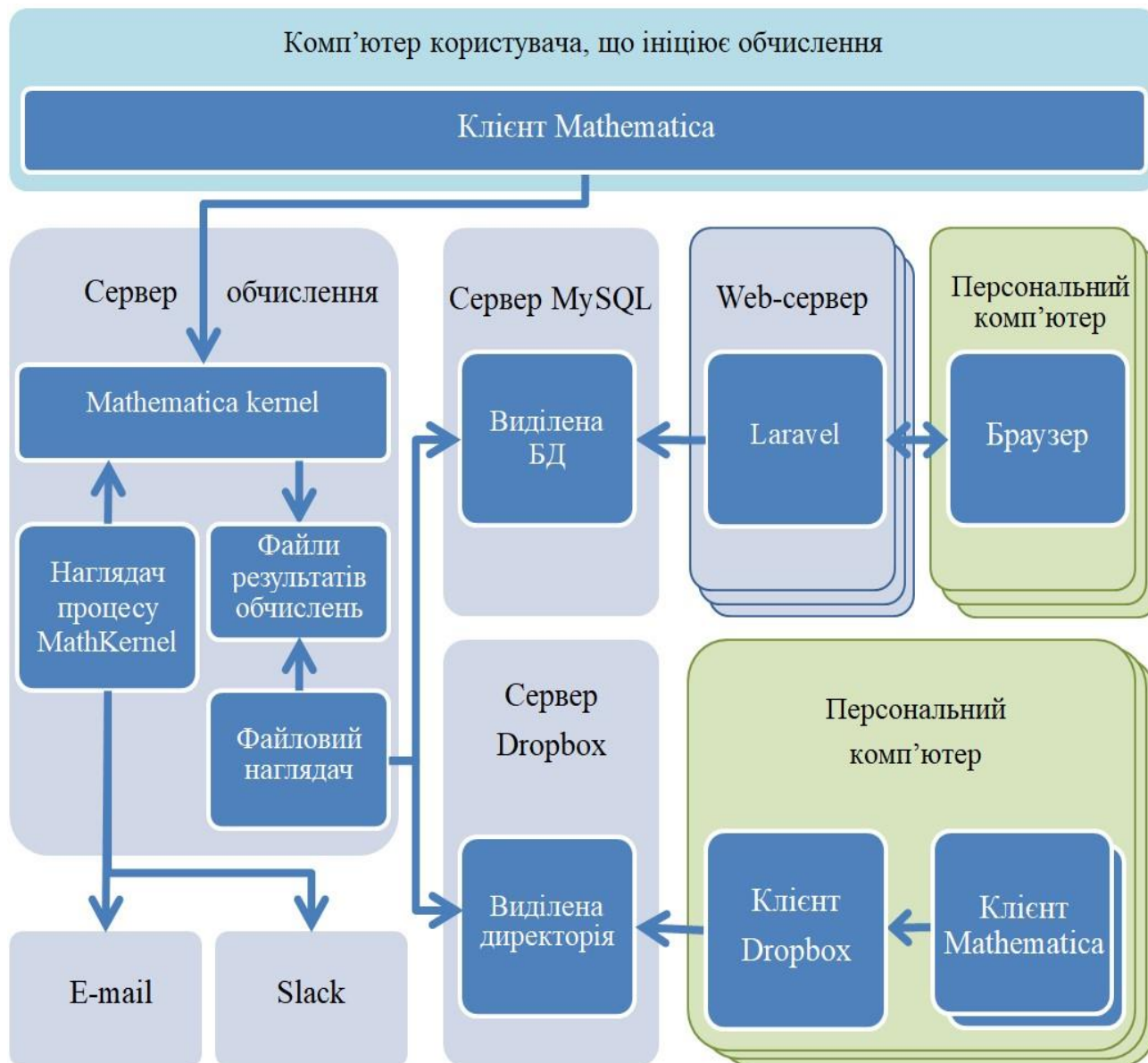


Рис. 1. Декомпозиція системи автоматизованих обчислень на структурні елементи

Розглянемо далі кожний елемент системи докладніше.

1. Ініціатор обчислень

Має мережевий доступ через ssh-канал до центру обчислень із використанням ключів асиметричного шифрування, що гарантують повну безпечність передачі команд та прийняття результатів від сервера обчислень.

Через те що під час роботи програма Mathematica виводить у свій документ результати виконання різних операцій: обчислення, протокол роботи, графіки, проміжні значення змінних і таке інше, – щоб уникнути їх втрати у разі збою включене автоматичне збереження файлу одним з двох способів:

- за допомогою установки опції документа NotebookAutoSave [13] у значення True наступною конструкцією: `SetOptions[EvaluationNotebook[], NotebookAutoSave -> True]`,

- через ручне збереження за допомогою NotebookSave [13] у місцях програми обраних користувачем.

Перший спосіб зберігає файл після кожної зміни та підходить для випадків невеликої частоти зміни документа. Якщо ж програма генерує багато виводу у документ, то такий підхід може істотно її сповільнити. У такому разі рекомендується використовувати другий спосіб, та зберігати документ кожний раз після виводу важливого пакету результатів обчислень, наприклад, після кожної ітерації або кроку алгоритму.

2. Сервер обчислень

Надає клієнтам захищений мережевий доступ через ssh-канал, що використовує аутентифікацію виключно за ключами асиметричного шифрування. Це запобігає випадкам втрати або загублення паролів доступу, або неправомірному доступу через їх крадіжку.

Впровадження протоколювання (логування) у програму переслідує дві мети: виявити поточний крок

виконання алгоритму й визначити час і місце помилки у випадку збою в роботі або при аварійному завершенні. Під зовнішнім протоколюванням слід розуміти запис у таке сховище даних, що не залежить від програми обчислення: сторонній файл, база даних, Elasticsearch, інший комп'ютер і таке інше.

Для побудови такого механізму обраний варіант запису у файл через те, що такий спосіб достатньо надійний, підтримує легке дублювання, та не потребує необґрунтованого ускладнення системи. У назву файлу включено дату й час із точністю до годин, на момент здійснення запису: `:"log_" + дата у форматі "Рік.Місяць.День_Години" + розширення "txt"`. Це означає, що кожен час створюється новий лог-файл, та запис переноситься до нього. Така назва унікальна у рамках директорії, а файли із назвою у такому форматі дати легко сортувати та шукати серед них потрібний. Із функціональної точки зору це дає можливість кожен час видаляти або переносити в архів старий лог-файл для аналізу та наочно бачити, що програма працює, дивлячись у директорію протоколювання роботи.

Лог-файл відкривається програмою для дозапису у режимі спільного доступу (read shared mode), що дає змогу читати цей файл сторонніми додатками для аналізу під час запису. У цьому файлі зберігаються:

- час події з точністю до мілісекунд,
- назва події,
- інформація про успішність її запуску або завершення.

Потрібно звернути увагу на важливість запису про успішність кожного етапу події окремим рядком, а не пакетний запис про всю подію цілком – у протилежному випадку при серйозному збої рядок про всю подію може зовсім не записатися у лог-файл, тим самим розширюючи область пошуку помилки для розробника.

Текст кодується у UTF8, тому що це дозволяє зберігати разом із звичайними символами також спеціальні, що часто полегшує читання для науковця. Програма у Mathematica налаштована таким чином, що додає кожний рядок у форматі StandardForm, випереджаючи його датою й часом у форматі "дд-мм-рррр чч:хх:сс.мс". Формат StandardForm гарантує, що усі символи будуть друкованими та серед них не буде службових даних, що відносяться до внутрішнього формату Wolfram Mathematica.

Збереження результатів виконання кожної ітерації переслідує дві мети: захистити вже отримані результати обчислень від збоїв при обчисленні наступних, та створити можливість ручної зупинки й запуску програми у довільний момент часу, а також поновлення її роботи після аварійного збою. Що також підвищує відмовостійкість програми в цілому. У загальному випадку, зберігання результатів у зовнішній пам'яті також дозволяє без труднощів переносити результати (частково або всі) і продовжувати виконання програми на інших комп'ютерах.

Експортування проводиться зразу двома шляхами: функцією DumpSave [13] – у файл дампу

для найшвидшого зворотного завантаження у Mathematica, та функцією Export [13] – у текстовому форматі для обробки сторонніми додатками. Результати кожної ітерації записуються до окремого файлу, а не додаються у один (у кінець). У останньому випадку збій під час запису до файлу зробить його не придатним до використання, крім того файл дампу може бути створений лише один раз в силу своєї природи та не може бути модифікований, лише перезаписаний знову. Треба зазначити, що за потреби експортувати можна також і в інші платформо-незалежні формати, що підтримуються функцією Export.

Механізм збереження повинен заздалегідь підтримувати автоматичне розділення за значеннями вхідних параметрів обчислень, щоб уникнути плутанини. Це досягається за допомогою створення виділеної директорії для конкретної задачі, назва якої містить значення параметрів обчислень, де перебувають папки з лог-файлами, результатами ітерацій, загальними результатами, експортованими графіками й тому подібне. Крім того рекомендується надати користувачеві можливість зсередини програми додавати в кінець назви директорії коротку строку деякої службової інформації: про тип майбутніх обчислень, додаткові характеристики і таке інше. З практики [15], найменування вигляду "Param1Val1_Param2Val2_..._AdditionalStr" зручне для класифікації результатів обчислень однією програмою однієї задачі із різними вхідними параметрами.

Миттєве сповіщення про збій у процесі обчислення. Виконання програми у ядрі Mathematica часто перевищує декілька годин та навіть днів. Під час обчислень ядро може аварійно завершитись: через нестачу оперативної пам'яті, внутрішнього збою у алгоритмах Mathematica, збою у роботі мережі та іншого. У такому разі зручно мати механізм автоматичного сповіщення про таку ситуацію. Контролювати це зсередини матпакету неможливо, бо ядро – основний його центр обробки команд користувача, і майже всі операції проходять за його участі. Тому розроблено сторонній додаток на базі bash-скриптів автоматичного реагування на запуск та закриття ядра Mathematica – процесу WolframKernel. Його ідея складається з додавання фонові задачі, що:

- при запуску ядра однократно створює файл `"~/mathkernel.run"`, що містить PID процесу WolframKernel отриманий командою `"pgrep WolframKernel"`, та відправляє сповіщення на e-mail та у Slack [16],
- при завершенні усіх ядер видаляє цей файл та відправляє інше сповіщення на e-mail та у Slack.

Для відправки e-mail використовується Mailutils 3.7, для відправлення сповіщення у Slack – створений додаток у Slack та розроблений bash-скрипт для роботи із його веб-API.

Важливим зауваженням є те, що до одного серверу одночасно підключається тільки один клієнт-ініціатор, програма якого у процесі виконання може запускати декілька ядер для паралельних обчислень. Тому сигналом аварії служить закриття

усіх процесів ядра, а PID використовується для однозначного визначення кореневого процесу.

Очистка займаних ресурсів після закінчення програми. Наприкінці роботи програми закривається ядро Mathematica, щоб звільнити оперативну пам'ять для інших додатків на сервері. Для цього служить функція Quit [13], що підтримує повернення коду завершення оточенню через свій аргумент. При закритті ядра губляться всі визначення функцій і даних, які в ньому були, і не були збережені в зовнішній пам'яті чи поточному документі Mathematica.

3. Канал зв'язку із клієнтами, що обробляють результати

Для аналізу та обробки результатів обчислення ітерацій додатками на сторонніх комп'ютерах файли цих результатів повинні зберігатися на:

- носії, доступному зовні (з мережі LAN тощо),
- іншому комп'ютері у мережі LAN, зовнішньому сервісі тощо.

Перший варіант є більш стабільним, надійним та прозорим для додатка, бо логічно та фізично він не відрізняється від зберігання у звичайній, недоступній зовні, директорії. Другий – більше підходить при великій кількості звернень до результатів обчислень із сторонніх комп'ютерів. Логічно він також не відрізняється для додатка, але фізично значно складніший і через це більш ненадійний. Але він дає можливість зберігши один раз результати обчислення зовні, обробляти їх різними аналітичними та статистичними програмними комплексами, не турбуючи обчислювальний додаток та розвантажити сервер від звернень із мережі. Також останній варіант дає більшу гарантію збереження даних при фізичній аварії.

При побудові системи використаний гібридний підхід – дані та протоколи роботи зберігаються локально без доступу із зовнішніх систем. Спеціально створений додаток на базі bash-скриптів:

- дублює їх у сховище Dropbox через web-API,
- додає їх вміст після розбору (парсингу) до бази даних MySQL на сторонньому сервері.

Автоматичне виявлення появи нових файлів засноване на службовому системному додатку inotifywait [17], що при генеруванні системою події закінчення запису до файлу та його закриття запускає вище описану обробку. Ключова частина скрипту, що відповідає за обробку події, наступна: `inotifywait -m -r -q --format "%f" -e close_write "$DIR" | while read FILE, де "$DIR" – шлях до директорії для наглядання.`

Коли обчислення однієї ітерації займає значний час у додатку є можливість включити відправлення сповіщення у Slack після кожного додавання файлу у директорію результатів ітерацій, що дозволяє кожній зацікавленій особі бачити прогрес розрахунку. Але таку можливість не рекомендується використовувати, якщо час обчислення ітерації менше 15 хвилин, бо повідомлення приходять дуже часто та не мають сенсу.

Переваги використання сервісу Dropbox для дублювання результатів обчислень та файлів

протоколу роботи у порівнянні із використанням свого власного серверу для розробленої системи наступні:

- синхронізація із пристроєм користувача проходить автоматично у фоні та майже миттєво, при цьому формується локальна незалежна копія даних, що завжди зберігається, а звернення до неї не навантажують сервер обчислення та сервер Dropbox,
- для доступу потрібен лише додаток Dropbox або браузер на смартфоні, планшеті, ноутбукі, комп'ютері тощо.

4. Клієнти обчислювальної системи, що обробляють дані

Засновані на використанні каналів зв'язку, що описані вище, і як наслідок, здійснюють аналіз результатів обчислень кожної ітерації одним з двох шляхів, кожен з яких має як свої переваги так і недоліки.

Перший – на персональному комп'ютері із ОС Windows 10 із підключеною директорією Dropbox, через однойменний клієнт, у якості віддаленої директорії та Wolfram Mathematica 8, що при появі нових файлів у ній читає їх та обробляє. Безпека базується на вбудованих можливостях клієнту Dropbox.

Його переваги очевидні: доступ до всіх вбудованих аналітичних функцій математичного процесору, різноманітні засоби обробки та аналізу, зміна алгоритмів обробки та аналізу може бути проведена індивідуально та у будь-який час за побажанням користувача, можливість обробки будь-яким іншим програмним забезпеченням за побажанням користувача.

Недоліки: потрібне стороннє програмне забезпечення, а доступ до результатів такого аналізу є тільки на тому комп'ютері, де вони отримані.

Другий – розроблений веб-додаток, заснований на Laravel (версія мови програмування PHP 7.4), що оброблює запити на отримання веб-сторінок через браузер кінцевого користувача, та відповідає на API-запити для отримання та автоматичного оновлення даних через механізм ajax. Аутентифікація заснована на класичному підході: вхід за логіном та паролем, обробка із використанням cookie та сесій, протокол з'єднання HTTPS.

Його переваги: доступ є у кожного користувача, що має доступ, із будь-якої частини світу через будь-який пристрій, що має встановлений браузер, також відсутнє навантаження на клієнт на обробку даних – увесь аналіз проходить на веб-сервері.

Недоліки: дуже малий спектр аналітичних операцій, якими можливо оброблювати дані, у порівнянні із спеціалізованим математичним забезпеченням, а для обробки можуть бути використані тільки заздалегідь запрограмовані на веб-сервері алгоритми аналізу.

Окремо треба зупинитися на використаних технологіях та відмітити, що мова програмування PHP, фреймворк Laravel, СУБД MySQL обрані тільки через те, що під час розробки системи вони виявилися найбільш придатними за швидкістю розробки та

розгортання. Такого ж самого результату можна досягти будь-якими іншими аналогічними засобами.

Демонстрація роботи

Для демонстрації роботи побудованої системи обраний розрахунок задачі обчислення розповсюдження імпульсу Ейрі у одномірному плоскошаруватому середовищі методом апроксимуючих функцій [9]. Увесь процес обчислення ведеться для відрізка нормованого часу $t=[0, 10]$ та

нормованого простору $x=[0, 1]$, кроком моделювання $h=0.02$.

Розглянемо обчислення ітерації №17 для відрізка часу $t=[8.5, 9.0]$. Це означає, що на попередніх ітераціях для відрізка часу $t=[0, 8.5]$ результати вже отримані, збережені локально та віддалено. Рисунок, що наведені нижче отримані у однаковий час.

На клієнті-ініціаторі обчислення – вивід у документі Mathematica показаний на рис. 2. Такі ж саме строки записуються до файлу протоколу роботи.

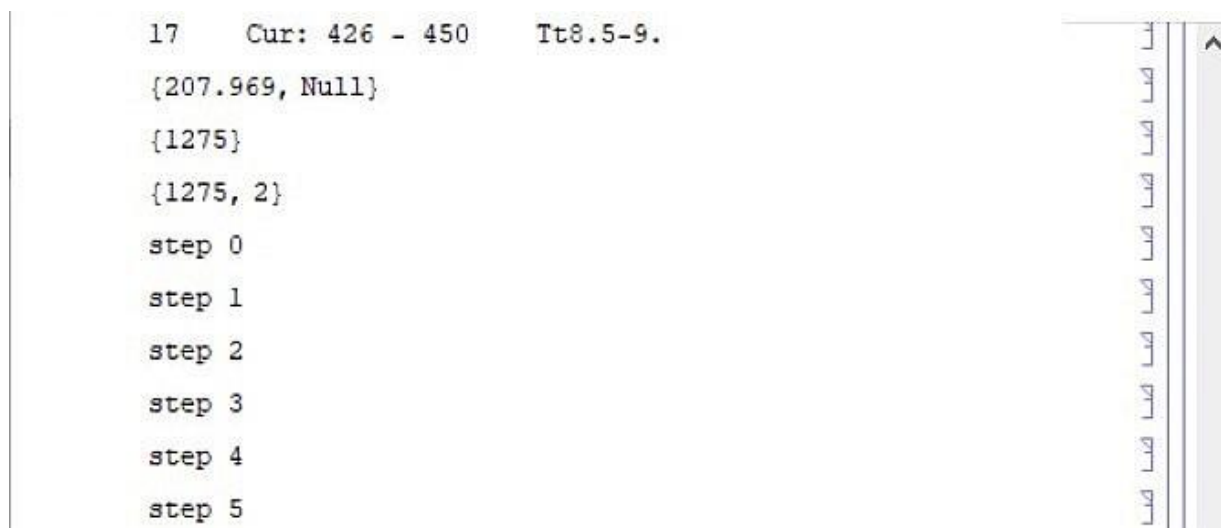


Рис. 2. Вивід у документі Mathematica на клієнті-ініціаторі обчислення

На сервері обчислення, що розташований у хмарному середовищі, у ssh-консолі у той же час буде наступний вивід, показаний на рис. 3. Видно, як файловий наглядчик отримує сповіщення про створення нового файлу результатів ітерації, що має

назву вигляду "B1_Airy_NL_Tt*.*.txt", де зірочки – це цифри, та автоматично завантажує їх до сервісу Dropbox разом із додаванням записів до бази даних. Аналогічним чином проходить обробка створення та зміни файлів протоколу роботи.

New iteration results found B1_Airy_NL_Tt6.5-7.0.txt:

> Dropbox: Uploading "~/math/2d/Airy/Airy_NL/ee1_ee13_h0.02_Tt10_T00_X00/B1_Airy_NL_Tt6.5-7.0.txt" to "~/math/B1_Airy_NL_Tt6.5-7.0.txt"... DONE

> MySQL: Adding records for time interval '6.5 - 7.0'... DONE

New iteration results found B1_Airy_NL_Tt7.0-7.5.txt:

> Dropbox: Uploading "~/math/2d/Airy/Airy_NL/ee1_ee13_h0.02_Tt10_T00_X00/B1_Airy_NL_Tt7.0-7.5.txt" to "~/math/B1_Airy_NL_Tt7.0-7.5.txt"... DONE

> MySQL: Adding records for time interval '7.0 - 7.5'... DONE

New iteration results found B1_Airy_NL_Tt7.5-8.0.txt:

> Dropbox: Uploading "~/math/2d/Airy/Airy_NL/ee1_ee13_h0.02_Tt10_T00_X00/B1_Airy_NL_Tt7.5-8.0.txt" to "~/math/B1_Airy_NL_Tt7.5-8.0.txt"... DONE

> MySQL: Adding records for time interval '7.5 - 8.0'... DONE

New iteration results found B1_Airy_NL_Tt8.0-8.5.txt:

> Dropbox: Uploading "~/math/2d/Airy/Airy_NL/ee1_ee13_h0.02_Tt10_T00_X00/B1_Airy_NL_Tt8.0-8.5.txt" to "~/math/B1_Airy_NL_Tt8.0-8.5.txt"... DONE

> MySQL: Adding records for time interval '8.0 - 8.5'... DONE

Рис. 3. Вивід у ssh-консолі файлового наглядчика на сервері обчислень

Один з клієнтів, що оброблює результати обчислень має наступний фрагмент виводу у документі Mathematica, що наведений на рис. 4 – за допомогою розробленої функції будується графік

вихідної хвилі (суцільна лінія) та отриманої при розрахунках (пунктирна лінія) на повній осі моделювання $t=[0,10]$.

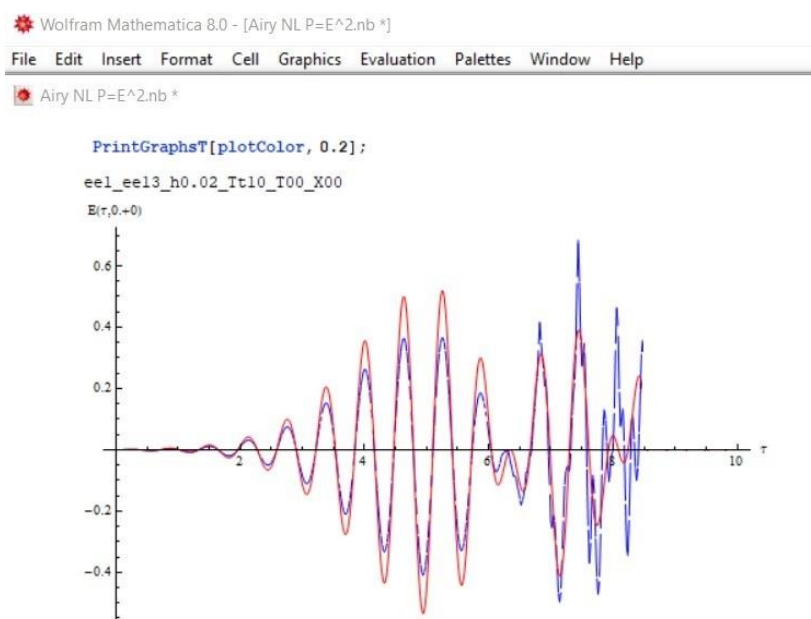


Рис. 4. Обробка результатів на клієнті у математичному процесорі Mathematica

Фрагмент виводу веб-клієнту, отриманий у браузері комп'ютера, наведений нижче на рис. 5: зверху – значення параметрів моделювання, ліворуч –

протокол роботи, праворуч – графік модельованої функції.

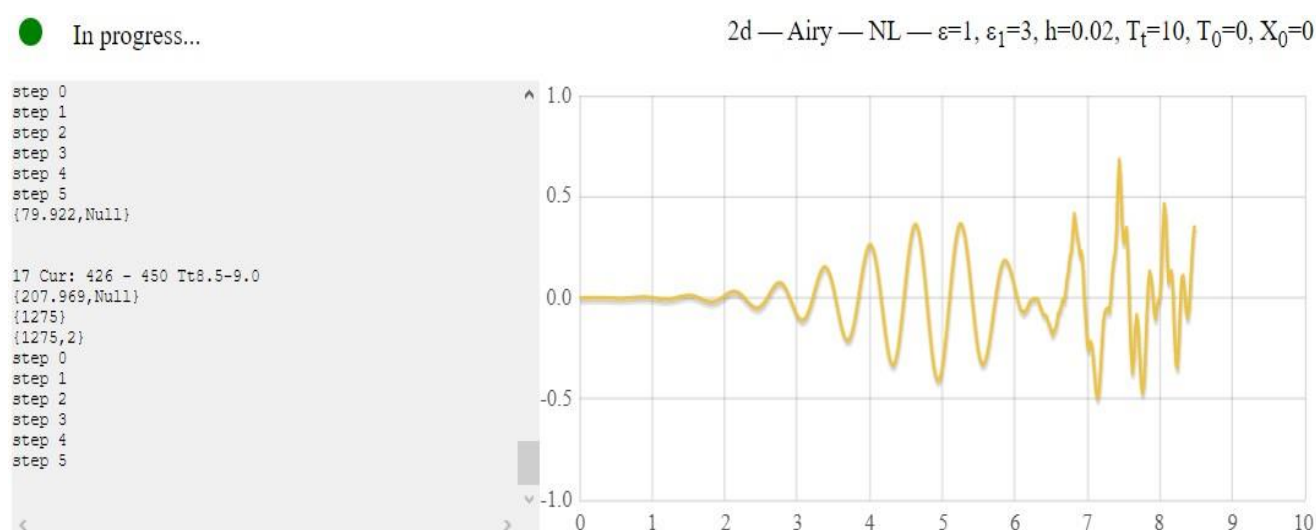


Рис. 5. Вивід оброблених на web-сервері результатів та протоколу роботи у браузері на клієнті

Перспективи подальшого розвитку системи

Перспективами подальшого розвитку розробленої системи бачаться наступні програмні вдосконалення. Підвищення безпеки збереження отриманих результатів, що відправляються на зовнішнє сховище у Dropbox, через попереднє шифрування асиметричними ключами. Для більш ретельного моніторингу роботи ядра Mathematica на сервері обчислень використовувати такий параметр як навантаження на центральний процесор.

Висновки

У роботі розроблені та обґрунтовані практичні основи для реалізації системи розподілених

автоматизованих обчислень та аналізу на базі хмарної інфраструктури, що складається з: клієнта-ініціатора обчислень, центра обчислень на базі ядра математичного пакету Wolfram Mathematica, каналів зв'язку із клієнтами, що обробляють результати, із використанням файлового сервісу Dropbox, серверів бази даних та веб-серверу; та клієнтів, що аналізують та обробляють результати за допомогою матпакету Mathematica або інтегрованого середовища у браузері комп'ютера або смартфона.

Підходи, описані у статті, до формування системи дозволили, по-перше, підвищити надійність збереження результатів обчислення через їх потрійне дублювання: локальні файли на сервері обчислень, їх копії завантажені до сервісу Dropbox та записи на сторонній сервер бази даних. По-друге, аналізувати та

обробляти дані різними засобами та на різних серверах, що мають доступ до директорії у сервісі Dropbox або бази даних, та зробити паралельними всі процеси обробки результатів на кінцевих клієнтах, без побудовання черги за пріоритетом та очікування. По-третє, налагодити комфортний для користувача дистанційний контроль процесу обчислень та реагування на збої в роботі ядра, та наглядати за прогресом та обробкою результатів через будь-який смартфон або планшет. Через ітераційний алгоритм програми стало можливим переривати обчислення у середині, якщо програма дає вочевидь невірні результати, не очікуючи її закінчення, що може економити від годин то тижнів в залежності від складності задачі.

Економічна вигода від використання описаної системи досягається завдяки максимальній автоматизації, що призводить до підвищення якості праці дослідників через мінімізацію невикористаного робочого часу. Завдяки використанню хмарних технологій собівартість володіння інфраструктурою системи є значно меншою у порівнянні із реальною технікою та зведена до часу реального її використання, а час та витрати на її модифікацію найменші. Найбільшу економію коштів описана система дасть при нерегулярному використанні або у разі частоті зміни задач, що спричиняють істотну зміну конфігурації інфраструктури.

References

1. Jianhua Li, Jingyuan Wang, Yichao Teng, Zhiyong Xu & Jun Cheng (2020), "Broadband supercontinuum generation based on filled structural photonic crystal fibers with low incident optical power", *Optical and Quantum Electronics*, Vol. 52, No. 447. DOI: <https://doi.org/10.1007/s11082-020-02570-8>
2. Yingying Sun, Jin Zeng, Yejin Li, Peng Rao & Tingliang Hu (2020), "Compensation method for the image motion of a rotary-scan space camera", *Optical and Quantum Electronics*, Vol. 52, No. 502. DOI: <https://doi.org/10.1007/s11082-020-02624-x>
3. Linghong Jiang, Guoxu Zhang, Yiyang Zhang, Jinhong Zou, Chao Wang, Yuefeng Qi & Lantian Hou (2020), "A miniaturized and high-bandwidth polarization filter based on a plasmonic and liquid crystal offset core photonic crystal fiber", *Optical and Quantum Electronics*, Vol. 52, No. 409. DOI: <https://doi.org/10.1007/s11082-020-02515-1>
4. Raisa Mamtaz, Kawsar Ahmed, Bikash Kumar Paul, Md. Aslam Mollah, Mst. Nargis Aktar, Muhammad Shahin Uddin & Vigneswaran Dhasarathan (2020), "Design and FEM analysis of pentagonal photonic crystal fiber for highly non-linear applications", *Optical and Quantum Electronics*, Vol. 52, No. 455. DOI: <https://doi.org/10.1007/s11082-020-02562-8>
5. Zigang Li, Lulu Qiao, Jun Jiang, Ling Hong & Jian-Qiao Sun (2020), "Global dynamic analysis of the North Pacific Ocean by data-driven generalized cell mapping method", *International Journal of Dynamics and Control*, Vol. 8, P. 1141–1146. DOI: <https://doi.org/10.1007/s40435-020-00678-z>
6. Mahak Vij, Harsh Yadav, Nikita Vashistha, Manju Kumari, Hemant Kumar Verma, Prashant Kumar & K. K. Maurya (2020), "Crystal structure, Hirshfeld surface analysis and thermal behavior of diisopropylammonium succinate, a novel third-order nonlinear optical crystal", *Journal of Materials Science*, Vol. 55, P. 16900–16913. DOI: <https://doi.org/10.1007/s10853-020-05181-6>
7. Xuehong Ren, Shaopu Yang, Wenzhao Zhao & Guilin Wen (2020), "A crystal plasticity-based constitutive model for ratchetting of cyclic hardening polycrystalline metals", *International Journal of Dynamics and Control*, Vol. 8, P. 1161–1168. DOI: <https://doi.org/10.1007/s40435-020-00668-1>
8. Zolotariov, D., Nerukh, A. (2011), "Extension of the approximation functions method for 2d nonlinear Volterra integral equations", *Applied Radio Electronics*, Vol. 10, No. 1, P. 39–44.
9. Nerukh, A. G., Zolotariov, D. A., Nerukh, D. A. (2012), "Properties of decelerating non-diffractive electromagnetic Airy pulses", *Applied Radio Electronics*, Vol. 11, No. 1, P. 77–81.
10. Nerukh, A., Zolotariov, D. & Benson, T. (2015), "The approximating functions method for nonlinear Volterra integral equations", *Optical and Quantum Electronics*, Vol. 47, P. 2565–2575. DOI: <https://doi.org/10.1007/s11082-015-0141-2>
11. Nerukh, A., Zolotariov, D., Kuryzheva, O. & Benson T. (2016), "Dynamics of decelerating pulses at a dielectric layer", *Optical and Quantum Electronics*, Vol. 48, No. 89. DOI: <https://doi.org/10.1007/s11082-016-0386-4>
12. Wolfram (2020), "Wolfram Mathematica: Modern technical calculations", available at: <https://www.wolfram.com/mathematica/> (last accessed 12 November 2020).
13. Wolfram (2020), "Wolfram Language & System Documentation Center", available at: <https://reference.wolfram.com/language/> (last accessed 12 November 2020).
14. Laravel (2020), "Laravel – The PHP Framework for Web Artisans", available at: <https://laravel.com/docs/6.x/> (last accessed 12 November 2020).
15. Zolotariov, D. A. (2020), *Automation and optimization of scientific and engineering calculations in Wolfram Mathematica*, Kharkiv : FOP Panov A. M. ISBN: 978-617-7859-36-8
16. Slack (2020), "Where work happens", available at: <https://slack.com/intl/en-en/> (last accessed 12 November 2020).
17. Linux (2020), "inotifywait(1) - Linux man page", available at: <https://linux.die.net/man/1/inotifywait> (last accessed 12 November 2020).

Надійшла (Received) 13.11.2020

Відомості про авторів / Сведения об авторах / About the Authors

Золотарьов Денис Олексійович – кандидат фізико-математичних наук, Харків, Україна, email: denis@zolotariov.org.ua, ORCID: <https://orcid.org/0000-0003-4907-7810>.

Золотарёв Денис Алексеевич – кандидат физико-математических наук, Харьков, Украина.

Zolotariov Denis – PhD (Physics and Mathematics Sciences), Kharkiv, Ukraine.

РАСПРЕДЕЛЕННАЯ СИСТЕМА АВТОМАТИЗИРОВАННЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ ОБЛАЧНОЙ ИНФРАСТРУКТУРЫ

Статья посвящена теоретическому исследованию и разработке распределенной системы автоматизированных вычислений и анализа на базе облачной инфраструктуры. **Предметом** исследования являются теоретические и практические основы построения систем автоматизированных вычислений и анализа, основанных на клиент-серверной архитектуре в распределенной инфраструктуре на базе облачных технологий. **Целью** статьи является разработка и обоснование практических рекомендаций к формированию инфраструктуры системы автоматизированных вычислений, выбору ее составных элементов и их компонентов. **Задача** работы: выявить необходимые структурные элементы системы автоматизированных вычислений и провести для каждого из них анализ составляющих компонентов и функциональной нагрузки, поставить конкретные задачи для построения каждого из них и обосновать выбор инструментов для их решения. В ходе исследования использованы **методы** системного анализа для декомпозиции сложной системы на элементы и каждого элемента на функциональные компоненты. В **результате** исследования установлено, что инфраструктура системы должна состоять из: единых инициатора вычислений и центра вычислений, каналов связи с клиентами, и конечными клиентами, которые обрабатывают результаты. Вся система должна работать в реальном или близком к реальному времени, что должно быть достигнуто через обработку событий операционной системы и внешних сервисов. Каналы связи должны быть максимально универсальными и предоставлять широкие возможности клиентам для доступа к данным. Конечные клиенты должны быть двух типов: полноценные для гибкой и индивидуальной обработки данных, и упрощенные для интегрированной среды обобщенной обработки данных в мобильном или десктопном браузере. По полученным рекомендациям разработан один из вариантов такой системы, показаны принципы его работы и приведены результаты. **Выводы.** Разработаны теоретические и практические основы для реализации системы распределенных автоматизированных вычислений и анализа на базе облачных технологий. Показано повышение надежности хранения результатов моделирования в такой системе через их тройное дублирование благодаря: локальному хранилищу, использованию внешних базы данных и файлового сервиса. Предоставлены доводы по повышению удобства и гибкости в обработке результатов благодаря возможности использования сторонних аналитических приложений, которые поддерживают загрузку данных из таких источников. Показана экономическая выгода от использования описанной системы. Показаны будущие пути ее совершенствования.

Ключевые слова: облачные технологии; распределенная инфраструктура; автоматизированные вычисления; экономия ресурсов и денег; итерационные алгоритмы; Mathematica.

THE DISTRIBUTED SYSTEM OF AUTOMATED COMPUTING BASED ON CLOUD INFRASTRUCTURE

The article is devoted to theoretical research and development of a distributed system of automated calculations and analysis based on cloud infrastructure. The **subject** of the research is theoretical and practical principles of building automated calculation and analysis systems based on client-server architecture in a distributed infrastructure based on cloud technologies. The **purpose** of the article is to develop and substantiate practical recommendations for the formation of automated computing system infrastructure, the choice of its elements and their components. **Tasks** of the study is to identify the necessary structural elements of the system of automated calculations and provide for each of them an analysis of components and functional load, set specific tasks for the construction of each of them and substantiate the choice of tools for their solution. The **methods** of system analysis to decompose a complex system into elements and each element into functional components were used during the study. The study found that the system infrastructure should consist of: a single computing initiator and computing center, customer communication channels, and end customers who process the **results**. The entire system must run in real time or near real time mode that must be achieved through the handling of operating system events and external services events. Communication channels should be as versatile as possible and provide customers with ample opportunities to access data. The end customers should be of two types: powerful for flexible and individual data processing, and simplified for the integrated environment of generalized data processing in mobile or desktop browsers. According to the received recommendations one of variants of such system is developed, principles of its work and results are shown. **Conclusions.** The theoretical and practical foundations for the implementation of a system of distributed automated computing and analysis based on cloud technologies have been developed. An increase in the reliability of storing the simulation results in such a system through their triple duplication is shown due to: local storage, using of external databases and external file service. Arguments are given to increase the convenience and flexibility in processing the results due to the possibility of using third-party analytical applications that support getting data from such sources. The economic benefits of using the described system are shown. The future ways of its improvement are given.

Keywords: cloud technologies; distributed infrastructure; automated cloud calculations; saving resources and funds; iterative algorithms; Mathematica.

Бібліографічні описи / Bibliographic descriptions

Золотарьов Д. О. Розподілена система автоматизованих обчислень на базі хмарної інфраструктури. *Сучасний стан наукових досліджень та технологій в промисловості*. 2020. № 4 (14). С. 47–55. DOI: <https://doi.org/10.30837/ITSSI.2020.14.047>

Zolotariov, D. (2020), "The distributed system of automated computing based on cloud infrastructure", *Innovative Technologies and Scientific Solutions for Industries*, No. 4 (14), P. 47–55. DOI: <https://doi.org/10.30837/ITSSI.2020.14.047>